In The United States Patent and Trademark Office
On Appeal From The Examiner To The Board
of Patent Appeals and Interferences

In re Application of:	Richard H. Harvey
Serial No.:	09/827,738
Filing Date:	April 6, 2001
Group Art Unit:	2162
Examiner:	Jean B. Fleurantin
Confirmation No.:	6701
Title:	DIRECTORY SEARCHING METHODS AND SYSTEM

MAIL STOP: APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

APPEAL BRIEF

Appellant has appealed to the Board of Patent Appeals and Interferences (the "*Board*") from the Notice of Panel Decision dated June 24, 2008, finally rejecting Claims 1-7 and 22-28 of Claims 1-28 pending in this Application. Appellant filed a Notice of Appeal and Pre-Appeal Brief Request for Review with the statutory fee of $510.00 on May 21, 2008. Appellant respectfully submits this Appeal Brief in response to the Notice of Panel Decision dated June 24, 2008.

Real Party in Interest

This Application is currently owned by Computer Associates Think, Inc. as indicated by:

an assignment recorded on 06/18/2001 from inventor Richard H. Harvey to Computer Associates Think, Inc., in the Assignment Records of the PTO at Reel 011905, Frame 0984 (3 pages).

Related Appeals and Interferences

To the knowledge of Appellant's counsel, there are no known interferences or judicial proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision regarding this Appeal.

Status of Claims

Claims 1-28 are pending with Claims 8-21 being withdrawn pursuant to a Response to an Election/Restriction Requirement submitted on November 30, 2007.

Claims 1-7 and 22-28 stand rejected pursuant to a Final Office Action dated February 21, 2008 (the "*Final Office Action*) and the Notice of Panel Decision from Pre-Appeal Brief Review dated June 24, 2008 (the "*Panel Decision*"). Specifically, Claims 1-7 and 22-28 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over C.M.R. Leung, "An object-oriented approach to directory systems," 1990, pages 736-740 ("*Leung*") in view of J. Rumbaugh et al., "Object-Oriented Modeling and Design," 1991, pages 366-396 ("*Rumbaugh*"). For the reasons discussed below, Appellant respectfully submits that these rejections are improper and should be reversed by the Board.

Accordingly, Appellant presents Claims 1-7 and 22-28 for Appeal. All pending claims are shown in Appendix A, attached hereto, along with an indication of the status of those claims.

Status of Amendments

All amendments submitted by Appellant have been entered by the Examiner prior to the mailing of the *Final Office Action*.

Summary of Claimed Subject Matter

The present application relates to storing and/or searching for data types using some form of indicia, such as one of the components contained in stored data, an identifier of stored data and/or a manipulation of stored data. This implementation facilitates the searching of complex data types by adding new search and/or attribute tables that store information relating to data entries, predetermined information considered to be of assistance or useful in searching for particular entries of a database, such as, by not limited to including component identifier (CID) information representing an individual component and/or component value identifier information (CVID) representing a multi-valued component or components used for searching. These new tables are referred to herein as subsearch tables and/or subattribute tables which serve to store components of values, and facilitate the searching of individual components and/or multi-valued components. However, such tables can be referenced with any name. (*Applicants' Specification*, Page 4, line 1 - page 7, line 15.)

Referring to Figure 1b, the table structure of logical design 1 and physical design 2 include search table 3, subsearch table 4, attribute table 5, and subattribute table 6 . . . The subsearch table 4 can be configured to include one or more components used to improve the speed and reliability of the search. For example, the subsearch table 4 can include a CID field 7 and a CVID field 8. The CID field 4 serves as a tool (or index) for searching components of data types, and the CVID field 8 serves as a tool (or index) for multi-valued component searching. Although the method according to the present application can be used with one or more of the above-identified tables, preferably, each table is provided so that there is a choice for particular search queries. (*Applicants' Specification*, Page 7, lines 11-22.)

The attribute table 5 describes or references information in search table 3, and the subattribute table 6 describes or references information in subsearch table 4. The subattribute table 6 has similar fields as the attribute table 5, but substitutes CID field 9 for AID field 10. The CID field 9 is used to identify one or more components in the subsearch table 4. (*Applicants' Specification*, Page 7, line 23 - page 8, line 3.)

The subsearch table 4 preferably stores information that improves searching performance or components of complex data types. Other components stored in the subsearch table can be those that improve the manageability of the database. In other words, it is not a requirement that every value in a data entry is included in the subsearch table. (*Applicants' Specification*, Page 8, lines 4-8.)

One way to improve the efficiency of such a search is to utilize a subsearch table and search one or more components associated with the stored data. In such instances the database may be able to use an index to the component in the string or structure thus avoiding a scan of a large number of values. An example of an SQL statement for such a search would be:

SELECT EID {other columns} FROM SUBSEARCH {other tables} WHERE {filter} AND CID=x

In this implementation, the base object and whole tree searches 11 would use the subsearch table 4, the one level search 12 would use DIT table 14 and subsearch table 4, and the subtree search 13 would use tree table 15 and subsearch table 43. In this example, the filter used would reference a component of the data stored, which permits the use of an index and results in faster more efficient searches. The index used in this example includes CID=x. (*Applicants' Specification*, Page 9, lines 10-23.)

Figure 4 illustrates the application of the present application to X.509 certificates. For the purposes of clarity only a small part of each table is illustrated. The certificate 20 is illustrated schematically, and for the purposes of illustration only shows information, such as the issuer at field 21, validity information at field 22, serial number at field 23, version number at field 24, and subject information (e.g., rick harvey) at field 25. For this example, the search table 3 is arranged in one row with spaces (preferably two) separating the normalized value of each component or field of the certificate 20. The search table also includes a normalized value representing the entire certificate. The subsearch table 4 is arranged with one or more rows (26, 27, 28, 29 and 30) where one row corresponds to one component or field in certificate 20. It should be noted however that the subsearch table 4 does not have to include every field identified in certificate 20. (*Applicants' Specification*, Page 10, line 16 - page 11, line 4.)

To illustrate an implementation for this embodiment, assume that a simple certificate consists of information similar to what a credit card holds, e.g., a serial number, an expiration date, and the cardholders name. This simple certificate has three components or fields, namely a number field, a date field and a string field. In this simplified example, the normalized value of certificate 20 that would be stored in the search table 3 (of the form in Figure 1b) is as follows:

(xx, yy, zz, "123456 20000806123000 RICK HARVEY)"

And the subsearch table 4 may store, for example, three rows - one for each component of the certificate. Each row of the subsearch table would be in the form of Figure 1b:

(xx, yy, zz, 0, 0, "123456")
(xx, yy, zz, 1, 0, "20000806123000")
(xx, yy, zz, 2, 0, "RICK HARVEY")

where xx, yy and zz are integers corresponding to fields in the particular table design, such as EID, AID and VID. (*Applicants' Specification*, Page 11, lines 5-20.)

A search for a certificate that was issued to "RICK HARVEY" that utilized the search table 3 may use the following SQL statement:

SELECT ... FROM ... SEARCH ... WHERE AID = 27 and NORM LIKE '%RICK HARVEY%'

where the general filter is "NORM LIKE '%RICK HARVEY%'". However, using search table 3 for such a search may be slow because each component of the string for each entry would have to be examined and the degree of certainty that the string being searched and retrieved is the desired entry. This is because the flattened representation has no boundaries as it is an unstructured text representation. (*Applicants' Specification*, Page 11, line 21 through Page 12, line 6.)

A search for a certificate that was issued to "RICK HARVEY" would be more efficient if subsearch table 4 were utilized when applying an exact or initial filter and a component identifier were added to the search argument. In this instance, the following SQL statement may be used:

SELECT ... FROM ... SUBSEARCH ... WHERE ....AID = 27 AND CID = 4 AND NORM = 'RICK HARVEY'

Where "NORM = 'RICK HARVEY'"- is the filter, AID = 27 is the attribute identifier for a certificate and CID = 4 is the component identifier for the subject. (*Applicants' Specification*, Page 12, lines 7-15.)

In the above example, a search with the component identifier (or index) CID=4 is used because it is known from the design of subsearch table 4 or from subattribute table 6, that index CID=4 is a string representing cardholders name. A query where the index CID is set to 4, and the filter is "NORM = 'RICK HARVEY'" should return Rick Harvey's certificate. This query is considered to be better because it can make use of an appropriate index making the search faster and increasing the degree of certainty that the search will find the correct certificate or certificates. It should be noted that the actual designation of characters/letter or numerals in the subsearch table design is arbitrary, and may be designed in whatever manner to suit the particular application. (*Applicants' Specification*, Page 12, lines 16-25.)

With regard to the independent claims currently under Appeal, Appellant provides the following concise explanation of the subject matter recited in the claim elements. For brevity, Appellant does not necessarily identify every portion of the Specification and drawings relevant to the recited claim elements. Additionally, this explanation should not be used to limit Appellant's claims but is intended to assist the Board in considering the Appeal of this Application.

For example, independent Claim 1 recites the following:

A method of arranging and searching for data in a database comprising:

creating a first table storing data comprising at least one data entry, the data entry comprising a plurality of data components, the first table comprising one row for each data entry (e.g., Figure 4, reference numerals 30 and 20-25; Page 10, line 16 through Page 11, line 20);

creating a second table storing the plurality of data components of the data entry of the first table, the second table comprising one row for each of the plurality of data components of the data entry of the first table (e.g., Figure 4, reference numerals 4 and 26-30; Page 10, line 16 through Page 11, line 20); and

searching the rows of the second table to identify a particular one of the plurality of data components (e.g., Figure 1b, reference numeral 4, lines; Figure 4b, reference numerals 4 and 26-30; Page 8, lines 4-8; Page 9, lines 10-23; Page 11, line 21 through Page 12, line 15); and

returning the given data entry from the first table that includes the particular one of the plurality of data components (e.g., Page 5, lines 13-17).

Independent Claim 22 recites the following:

A method of searching a database for given data entries, comprising:

determining a component of a given data entry of a first table, the given data entry of the first table comprising a plurality of data components (e.g., Figure 4, reference numerals 30 and 20-25; Page 10, line 16 through Page 11, line 20);

identifying a component identifier indicating a data type that is associated with the component of the first table (e.g., Figure 1b, reference numerals 7 and 9; Abstract; Page 4, lines 2-15; Page 7, line 15 through Page 8, line 8);

using the component identifier indicating the data type to execute one of an exact or initial matching on a column of a second table in order to locate the component in the second table, the second table comprising one row for each of the plurality of data components of the given data entry of the first table (e.g., Figure 1b, reference numeral 4, lines; Figure 4b, reference numerals 4 and 26-30; Page 8, lines 4-8; Page 9, lines 10-23; Page 11, line 21 through Page 12, line 15); and

returning the given data entry from the first table matching the component located (e.g., Page 5, lines 13-17).

Grounds of Rejection to be Reviewed on Appeal

Are Claims 1-7 and 22-28 unpatentable under 35 U.S.C. § 103(a) over C.M.R. Leung, "An object-oriented approach to directory systems," 1990, pages 736-740 ("*Leung*") in view of J. Rumbaugh et al., "Object-Oriented Modeling and Design," 1991, pages 366-396 ("*Rumbaugh*")?

Argument

Claims 1-7 and 22-28 are rejected as being unpatentable under 35 U.S.C. § 103(a) over the proposed *Leung-Rumbaugh* combination. For at least the following reasons, Appellants respectfully submit that these rejections are improper and should be reversed by the Board. Appellants address independent Claims 1 and 22 and dependent Claims 4-5 and 25-28 below.

I. Legal Standard for Obviousness

The question raised under 35 U.S.C. § 103 is whether the prior art taken as a whole would suggest the claimed invention taken as a whole to one of ordinary skill in the art at the time of the invention. One of the three basic criteria that must be established by an Examiner to establish a *prima facie* case of obviousness is that "the prior art reference (or references when combined) must teach or suggest ***all the claim limitations***." *See* M.P.E.P. § 706.02(j) citing *In re Vaeck*, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991) (emphasis added). "***All words*** in a claim must be considered in judging the patentability of that claim against the prior art." *See* M.P.E.P. § 2143.03 citing *In re Wilson*, 424 F.2d 1382, 1385 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970) (emphasis added).

In addition, even if all elements of a claim are disclosed in various prior art references, which is certainly not the case here as discussed below, the claimed invention taken as a whole still cannot be said to be obvious without some reason why one of ordinary skill at the time of the invention would have been prompted to modify the teachings of a reference or combine the teachings of multiple references to arrive at the claimed invention.

The controlling case law, rules, and guidelines repeatedly warn against using an applicant's disclosure as a blueprint to reconstruct the claimed invention. For example, the M.P.E.P. states, "The tendency to resort to 'hindsight' based upon applicant's disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art." M.P.E.P. § 2142.

The U.S. Supreme Court's decision in *KSR Int'l Co. v. Teleflex, Inc.* reiterated the requirement that Examiners provide an explanation as to why the claimed invention would have been obvious. *KSR Int'l Co. v. Teleflex, Inc.*, 127 S.Ct. 1727 (2007). The analysis regarding an apparent reason to combine the known elements in the fashion claimed in the patent at issue "should be made explicit." *KSR*, 127 S.Ct. at 1740-41. "Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *Id.* at 1741 (internal quotations omitted).

The new examination guidelines issued by the PTO in response to the *KSR* decision further emphasize the importance of an explicit, articulated reason why the claimed invention is obvious. Those guidelines state, in part, that "[t]he key to supporting any rejection under 35 U.S.C. 103 is the clear articulation of the reason(s) why the claimed invention would have been obvious. The Supreme Court in *KSR* noted that the analysis supporting a rejection under 35 U.S.C. 103 should be made explicit." *Examination Guidelines for Determining Obviousness Under 35 U.S.C. 103 in View of the Supreme Court Decision in KSR International Co. v. Teleflex Inc.*, 72 Fed. Reg. 57526, 57528-29 (Oct. 10, 2007) (internal citations omitted). The guidelines further describe a number of rationales that, in the PTO's view, can support a finding of obviousness. *Id.* at 57529-34. The guidelines set forth a number of particular findings of fact that must be made and explained by the Examiner to support a finding of obviousness based on one of those rationales. *See id.*

II. Claims 1-3 and 6-7 are Allowable over the proposed *Leung-Rumbaugh* Combination

Independent Claim 1 of the present application recites creating a first table and a second table. The second table is related to the first table in that the second table stores "the plurality of data components of the data entry of the first table" and includes "one row for each of the plurality of data components of the given data entry of the first table." Claim 1 further recites "searching the rows of the **second table** to identify a particular one of the plurality of data components." Finally, the given data entry is returned **from the first table** that includes the particular one of the plurality of data components" that was identified from searching the **second table**. Thus, Appellant's claim recites a method of searching a database

that requires the cooperation of two tables to search the second table to identify a particular data entry, and then return the given data entry from the first table that matches the component located in the second table. Appellants will demonstrate below that this combination of features and operations is not disclosed, taught, or suggested in the proposed *Leung-Rumbaugh* combination.

A. The proposed Leung-Rumbaugh combination does not disclose, teach, or suggest "*creating a second table storing the plurality of data components of the data entry of the first table*"

As an example of the deficiencies of the cited references, Appellant's contend that the proposed *Leung-Rumbaugh* combination does not disclose, teach, or suggest "creating a second table storing the plurality of data components of the data entry of the first table," as recited in Appellant's Claim 1. In the *Final Office Action*, the Examiner relies on *Leung* for disclosure of the first and second tables but on *Rumbaugh* for disclosure of "creating a second table storing the plurality of data components of the data entry of the first table." (*Final Office Action*, pages 6-8). However, Appellant respectfully contends that *Rumbaugh*, even in combination with *Leung*, does not disclose, teach, or suggest "creating a **second table** storing the plurality of data components of the data entry of the **first table**," as recited in Appellant's Claim 1.

Rumbaugh merely discloses "how to translate object models into DBMS code." (*Rumbaugh*, page 368, paragraph 2; page 374, paragraph 5). According to the process disclosed in *Rumbaugh*, "you should formulate object models for the external and conceptual schema. Then, you should translate each object model to ideal tables, that is, the table model." (*Rumbaugh*, page 373, paragraph 2). In the *Office Action*, the Examiner points to figure 17.12 as disclosing Appellant's claimed step. However, Figure 17.12 merely describes that an object model is translated into a table model, which is then translated into SQL code. (*Rumbaugh*, page 381, Figure 17.12; page 380, paragraphs 1-6). Similarly, Figure 17.13 illustrates an object model (for a qualified association), and Figure 17.14 illustrates the translation of the object model of Figure 17.13 into a table model. However, an object model is not analogous to Appellant's "first table" since an object model is not a table at all. Accordingly, since the table model of *Rumbaugh* is formed from an object model and an object model is not analogous to a table, the table model of *Rumbaugh* cannot be said to disclose, teach, or suggest "creating a

second table storing the plurality of data components of the data entry of the **first table**," as recited in Claim 1. At most *Rumbaugh* discloses creating a second table from an object model.

For these additional reasons, Appellants respectfully submit that Claim 1, together with Claims 2-3 and 6-7 that depend on Claim 1, are allowable over the proposed *Leung-Rumbaugh* combination.

B. The proposed Leung-Rumbaugh combination does not disclose, teach, or suggest "*the second table comprising one row for each of the plurality of data components of the given data entry of the first table*"

As another example of the deficiencies of the cited references, Appellant's contend that the proposed *Leung-Rumbaugh* combination does not disclose, teach, or suggest "the **second table** comprising one row for each of the plurality of data components of the given data entry of the **first table**," as recited in Appellant's Claim 1. In the *Final Office Action*, the Examiner relies on *Leung* for disclosure of the first and second tables and on *Rumbaugh* for disclosure of "creating a second table storing data components and having one row for each component of the data." (*Final Office Action*, pages 6-8). However, the disclosures of *Leung* and *Rumbaugh*, even in combination, do not disclose, teach, or suggest cooperation between first and second tables that results in "the **second table** comprising one row for each of the plurality of data components of the given data entry of the **first table**," as recited in Appellant's Claim 1.

Leung merely discloses an object-oriented database consisting of two objects "the DIT and ENTRY, stored as two relational tables," which are illustrated in Figure 6. (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). *Leung*'s DIT table "holds the information of the structure of the DIT." (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). In the DIT table, each entry occupies one row and contains "the system identifier of an object, that of its parent, and its RDN." (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). The ENTRY table, on the other hand, includes detailed information about each directory object. (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). In the ENTRY table, each row contains "the system identifier of [a directory] object, and an attribute value of an attribute type of the object in both normalized and raw forms." (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). Thus, each of the DIT and ENTRY tables store information for directory objects and include a

row for each object. For example, *Leung* provides the following Figure 6 to illustrate the DIT and ENTRY tables:

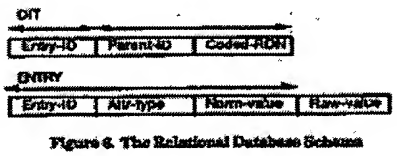


Figure 6. The Relational Database Schema

As can be seen from Figure 6, there is no disclosure in *Leung* of the ENTRY table "comprising one row for each of the plurality of data components of the given data entry" of the DIT, as is required by Appellant's Claim 1. Given the illustrated DIT table, were *Leung* to disclose Applicants' recited claim elements, the first row of the ENTRY table would include an entry for "Entry-ID", the second row of the ENTRY table would include an entry for "Parent-ID", and the third row of the ENTRY table would include an entry for "Coded-RDN". This is clearly not the case. At most, *Leung* can be said to disclose is that the first table includes one row for the data of the first table and that the second table includes one row for the data of the second table. Even though both tables may store data for common directory objects, *Leung* does not disclose, teach, or suggest that "the second table comprising one row for each of the plurality of data components of the given data entry of the first table," as recited in Claim 1.

Rumbaugh, which is relied upon specifically for disclosure of creating the second table, does not cure the deficiencies of *Leung*. *Rumbaugh* merely discloses "how to translate object models into DBMS code." (*Rumbaugh*, page 368, paragraph 2; page 374, paragraph 5). According to the process disclosed in *Rumbaugh*, "you should formulate object models for the external and conceptual schema. Then, you should translate each object model to ideal tables, that is, the table model." (*Rumbaugh*, page 373, paragraph 2). In the *Office Action*, the Examiner points to figure 17.12 as disclosing Appellant's claimed step. However, Figure 17.12 merely describes that an object model is translated into a table model, which is then translated into SQL code. (*Rumbaugh*, page 381, Figure 17.12; page 380, paragraphs 1-6). Similarly, Figure 17.13 illustrates an object model (for a qualified association), and Figure 17.14 illustrates the translation of the object model of Figure 17.13 into a table model. However, an object model is not analogous to Appellant's "first table" since an object model is not a table at

all. Accordingly, since the table model of *Rumbaugh* is formed from an object model and an object model is not analogous to a table, the table model of *Rumbaugh* cannot be said to be analogous to "the second table comprising one row for each of the plurality of data components of the given data entry of the **first table**," as recited in Claim 1.

For these additional reasons, Appellants respectfully submit that Claim 1, together with Claims 2-3 and 6-7 that depend on Claim 1, are allowable over the proposed *Leung-Rumbaugh* combination.

C. The proposed Leung-Rumbaugh combination does not disclose, teach, or suggest "*searching the rows of the second table to identify a particular one of the plurality of data components*" and "*returning the given data entry from the first table that includes the particular one of the plurality of data components*"

As another example of the deficiencies of the cited references, Appellant's contend that the proposed *Leung-Rumbaugh* combination does not disclose, teach, or suggest "searching the rows of the second table to identify a particular one of the plurality of data components" and "returning the given data entry from the first table that includes the particular one of the plurality of data components," as recited in Claim 1. In the *Final Office Action*, the Examiner identifies *Leung* as disclosing Appellant's recited steps. Appellant respectfully disagrees.

As discussed above, *Leung* discloses an object-oriented database consisting of two objects "the DIT and ENTRY, stored as two relational tables," which are illustrated in Figure 6. (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). Appellant respectfully submits, however, that *Leung* does not disclose, teach, or suggest performing operations on the DIT and ENTRY tables in a manner analogous to the steps of Appellant's claims. With respect to the DIT and ENTRY tables, *Leung* discloses a number of operations that may be performed on each. For example, operations that may be performed on the DIT include DitNavigate, DitAdd, DitRemove, DitChildren, DitParent, DitSubtree, and DitModifyRdn. (*Leung*, page 739, column 1, paragraph 2). Operations that may be performed on the ENTRY include Read, Add, Remove, Modify, ModifyRDN, Compare, GETRdn, and Search. (*Leung*, page 739, column 1, paragraph 2). Based on the descriptions of each of these operations, *Leung* indicates that the

operations performed on the DIT table are isolated to the DIT table and the operations performed on the ENTRY table are isolated to the ENTRY table.

There are no indications from *Leung* that the operations relating to the DIT and ENTRY tables are interrelated. For example, with respect to a "Search" operation performed on the ENTRY table, *Leung* discloses that the "Search" operation results in the return of "details of ENTRYs which satisfied the specified filter (search conditions) within the specified search domain (a list of system identifiers of objects to be searched)." (*Leung*, page 739, column 1, paragraph 2). As such, the "Search" operation to be performed on the ENTRY table as disclosed in *Leung* may not be used to identify an entry in the DIT table.

In a related passage, which is cited by the Examiner, *Leung* further discloses that the directory services supports "[d]irectory interrogation . . . composed of five abstract services: Read, Compare, List, Search, and Abandon." (*Leung*, page 737, column 2, paragraph 1). According to the system architecture described in *Leung*, a "front-end processor, is responsible for communicating with [Directory User Agents (DUAs)]." (*Leung*, page 737, column 2, paragraph 5). Thus, requests can be sent from a DUA to a [Directory System Agent (DSA)]" after an OSI session has been "established between the pair." (*Leung*, page 737, column 2, paragraph 5). The DSP, through a sub-system processor, performs the processing of the request and returns the results back to the requesting DUA. (*Leung*, page 737, column 2, paragraph 5). Accordingly, the cited portion of *Leung* merely discusses the front end processor by which a user can perform directory requests and that such requests may include a search function forwarded from a DUA to a DSP. The cited section does not change the operations of *Leung*, which are discussed in detail above. Certainly, the cited section does not alter the above discussed disclosure of *Leung*, which *Leung* indicates that the operations performed on the DIT table are isolated to the DIT table and the operations performed on the ENTRY table are isolated to the ENTRY table.

By contrast, Appellant's claim specifically recites "searching the rows of the **second table** to identify a particular one of the plurality of data components" and "returning the given data entry from the **first table** that includes the particular one of the plurality of data components." Thus, Appellant's method of searching a database requires the cooperation of

two tables to search the second table to identify a particular data component, and then return the given data entry from the first table that matches the component located in the second table. Appellant contends that the mere general disclosure in *Leung* of providing a search service for DIT and ENTRY tables is not analogous to Appellant's recited operations. Accordingly, *Leung* does not disclose, teach, or suggest "searching the rows of the **second table** to identify a particular one of the plurality of data components" and "returning the given data entry from the **first table** that includes the particular one of the plurality of data components," as recited in Claim 1.

For these additional reasons, Appellants respectfully submit that Claim 1, together with Claims 2-3 and 6-7 that depend on Claim 1, are allowable over the proposed *Leung-Rumbaugh* combination.

D. Conclusions with Respect to Claims 1-3 and 6-7

For at least these reasons, Appellants respectfully submit that the Examiner has not established a *prima facie* case of obviousness based on the proposed *Leung-Rumbaugh* combination with respect to independent Claim 1. Thus, for at least these reasons, Appellants submit that these rejections are improper and respectfully request that the Board reverse these rejections of independent Claim 1, together with Claims 2-3 and 6-7 that depend on Claim 1.

III. Claims 22-24 are Allowable over the proposed *Leung-Rumbaugh* Combination

Independent Claim 22 of the present application recites a first table and a second table. The second table is related to the first table in that the second table includes "one row for each of the plurality of data components of the given data entry of the first table." Claim 22 further recites "determining a component of a given data entry of a **first table**" and "identifying a component identifier indicating a data type that is associated with the component of the **first table**." The component identifier, which indicates the data type, is the used "to execute one of an exact or initial matching on a column of a **second table in order to locate the component in the second table**." Finally, Claim 22 then recites "returning the given data entry from the **first table** matching the component located." Thus, Appellant's claim recites a method of searching a database that requires the cooperation of two tables to

identify a component identifier in the first table, search the second table for the identifier, and then return the given data entry from the first table that matches the component located in the second table. Appellants will demonstrate below that this combination of features and operations is not disclosed, taught, or suggested in the proposed *Leung-Rumbaugh* combination.

A. *The proposed Leung-Rumbaugh combination does not disclose, teach, or suggest "the second table comprising one row for each of the plurality of data components of the given data entry of the first table"*

As another example of the deficiencies of the cited references, Appellant's contend that the proposed *Leung-Rumbaugh* combination does not disclose, teach, or suggest "the **second table** comprising one row for each of the plurality of data components of the given data entry of the **first table**," as recited in Appellant's Claim 22. In the *Final Office Action*, the Examiner relies on *Leung* for disclosure of the first and second tables and on *Rumbaugh* for disclosure of a second table "having one row for each component of the data." (*Final Office Action*, pages 6-8). However, the disclosures of *Leung* and *Rumbaugh*, even in combination, do not disclose, teach, or suggest cooperation between first and second tables that results in "the **second table** comprising one row for each of the plurality of data components of the given data entry of the **first table**," as recited in Appellant's Claim 22.

Leung merely discloses an object-oriented database consisting of two objects "the DIT and ENTRY, stored as two relational tables," which are illustrated in Figure 6. (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). *Leung*'s DIT table "holds the information of the structure of the DIT." (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). In the DIT table, each entry occupies one row and contains "the system identifier of an object, that of its parent, and its RDN." (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). The ENTRY table, on the other hand, includes detailed information about each directory object. (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). In the ENTRY table, each row contains "the system identifier of [a directory] object, and an attribute value of an attribute type of the object in both normalized and raw forms." (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). Thus, each of the DIT and ENTRY tables store information for directory objects and include a

row for each object. For example, *Leung* provides the following Figure 6 to illustrate the DIT and ENTRY tables:

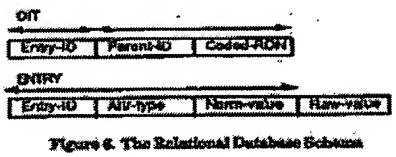


Figure 6. The Relational Database Schema

As can be seen from Figure 6, there is no disclosure in *Leung* of the ENTRY table "comprising one row for each of the plurality of data components of the given data entry" of the DIT, as is required by Appellant's Claim 22. Given the illustrated DIT table, were *Leung* to disclose Applicants' recited claim elements, the first row of the ENTRY table would include an entry for "Entry-ID", the second row of the ENTRY table would include an entry for "Parent-ID", and the third row of the ENTRY table would include an entry for "Coded-RDN". This is clearly not the case. At most, *Leung* can be said to disclose is that the first table includes one row for the data of the first table and that the second table includes one row for the data of the second table. Even though both tables may store data for common directory objects, *Leung* does not disclose, teach, or suggest that "the second table comprising one row for each of the plurality of data components of the given data entry of the first table," as recited in Claim 22.

Rumbaugh, which is relied upon specifically for disclosure of creating the second table, does not cure the deficiencies of *Leung*. *Rumbaugh* merely discloses "how to translate object models into DBMS code." (*Rumbaugh*, page 368, paragraph 2; page 374, paragraph 5). According to the process disclosed in *Rumbaugh*, "you should formulate object models for the external and conceptual schema. Then, you should translate each object model to ideal tables, that is, the table model." (*Rumbaugh*, page 373, paragraph 2). In the *Office Action*, the Examiner points to figure 17.12 as disclosing Appellant's claimed step. However, Figure 17.12 merely describes that an object model is translated into a table model, which is then translated into SQL code. (*Rumbaugh*, page 381, Figure 17.12; page 380, paragraphs 1-6). Similarly, Figure 17.13 illustrates an object model (for a qualified association), and Figure 17.14 illustrates the translation of the object model of Figure 17.13 into a table model. However, an object model is not analogous to Appellant's "first table" since an object model is not a table at

all. Accordingly, since the table model of *Rumbaugh* is formed from an object model and an object model is not analogous to a table, the table model of *Rumbaugh* cannot be said to be analogous to "the second table comprising one row for each of the plurality of data components of the given data entry of the **first table**," as recited in Claim 22.

For these additional reasons, Appellants respectfully submit that Claim 22, together with Claims 23-24 that depend on Claim 22, are allowable over the proposed *Leung-Rumbaugh* combination.

B. ***The proposed Leung-Rumbaugh combination does not disclose, teach, or suggest "using the component identifier indicating the data type to execute one of an exact or initial matching on a column of a second table in order to locate the component in the second table" and "returning the given data entry from the first table matching the component located"***

As another example of the deficiencies of the cited references, Appellant's contend that the proposed *Leung-Rumbaugh* combination does not disclose, teach, or suggest "using the component identifier indicating the data type to execute one of an exact or initial matching on a column of a second table in order to locate the component in the second table" and "returning the given data entry from the first table matching the component located," as recited in Claim 22. In the *Final Office Action*, the Examiner identifies *Leung* as disclosing Appellant's recited steps. (*Final Office Action*, page 8). Appellant respectfully disagrees.

As discussed above, *Leung* discloses an object-oriented database consisting of two objects "the DIT and ENTRY, stored as two relational tables," which are illustrated in Figure 6. (*Leung*, page 739, column 1, paragraph 1; *id.* at Figure 6). Appellant respectfully submits, however, that *Leung* does not disclose, teach, or suggest performing operations on the DIT and ENTRY tables in a manner analogous to the steps of Appellant's claims. With respect to the DIT and ENTRY tables, *Leung* discloses a number of operations that may be performed on each. For example, operations that may be performed on the DIT include DitNavigate, DitAdd, DitRemove, DitChildren, DitParent, DitSubtree, and DitModifyRdn. (*Leung*, page 739, column 1, paragraph 2). Operations that may be performed on the ENTRY include Read, Add, Remove, Modify, ModifyRDN, Compare, GETRdn, and Search. (*Leung*, page 739, column 1, paragraph 2). Based on the descriptions of each of these operations, *Leung* indicates that the

operations performed on the DIT table are isolated to the DIT table and the operations performed on the ENTRY table are isolated to the ENTRY table.

There are no indications from *Leung* that the operations relating to the DIT and ENTRY tables are interrelated. For example, with respect to a "Search" operation performed on the ENTRY table, *Leung* discloses that the "Search" operation results in the return of "details of ENTRYs which satisfied the specified filter (search conditions) within the specified search domain (a list of system identifiers of objects to be searched)." (*Leung*, page 739, column 1, paragraph 2). As such, the "Search" operation to be performed on the ENTRY table as disclosed in *Leung* may not be used to identify an entry in the DIT table.

In a related passage, which is cited by the Examiner, *Leung* further discloses that the directory services supports "[d]irectory interrogation . . . composed of five abstract services: Read, Compare, List, Search, and Abandon." (*Leung*, page 737, column 2, paragraph 1). According to the system architecture described in *Leung*, a "front-end processor, is responsible for communicating with [Directory User Agents (DUAs)]." (*Leung*, page 737, column 2, paragraph 5). Thus, requests can be sent from a DUA to a [Directory System Agent (DSA)]" after an OSI session has been "established between the pair." (*Leung*, page 737, column 2, paragraph 5). The DSP, through a sub-system processor, performs the processing of the request and returns the results back to the requesting DUA. (*Leung*, page 737, column 2, paragraph 5). Accordingly, the cited portion of *Leung* merely discusses the front end processor by which a user can perform directory requests and that such requests may include a search function forwarded from a DUA to a DSP. The cited section does not change the operations of *Leung*, which are discussed in detail above. Certainly, the cited section does not alter the above discussed disclosure of *Leung*, which *Leung* indicates that the operations performed on the DIT table are isolated to the DIT table and the operations performed on the ENTRY table are isolated to the ENTRY table.

By contrast, Appellant's claim specifically recites "using the component identifier indicating the data type to execute one of an exact or initial matching on a column of a **second table** in order to locate the component in the **second table**" and "returning the given data entry from the **first table** matching the component located." Thus, Appellant's method of searching

a database requires the cooperation of two tables to search the second table to identify a particular data component, and then return the given data entry from the first table that matches the component located in the second table. Appellant contends that the mere general disclosure in *Leung* of providing a search service for DIT and ENTRY tables is not analogous to Appellant's recited operations. Accordingly, *Leung* does not disclose, teach, or suggest "using the component identifier indicating the data type to execute one of an exact or initial matching on a column of a **second table** in order to locate the component in the **second table**" and "returning the given data entry from the **first table** matching the component located," as recited in Claim 22.

For at least these additional reasons, Appellants respectfully submit that Claim 22, together with Claims 23-24 that depend on Claim 22, are allowable over the proposed *Leung-Rumbaugh* combination.

C. Conclusions with Respect to Claims 22-24

For at least these reasons, Appellants respectfully submit that the Examiner has not established a *prima facie* case of obviousness based on the proposed *Leung-Rumbaugh* combination with respect to independent Claim 22. Thus, for at least these reasons, Appellants submit that these rejections are improper and respectfully request that the Board reverse these rejections of independent Claim 22, together with Claims 23-24 that depend on Claim 22.

IV. Claim 4 is Allowable over the proposed *Leung-Rumbaugh* Combination

Claim 4 is patentable at least because it depends on Claim 1, which Appellant has shown above to be allowable. Additionally, Claim 4 recites that "the data is or represents a X.509 certificate." Appellants will demonstrate below that this combination of features and operations is not disclosed, taught, or suggested in the proposed *Leung-Rumbaugh* combination.

A. *The proposed Leung-Rumbaugh combination does not disclose, teach, or suggest "the data is or represents a X.509 certificate"*

As an example of the deficiencies of the cited references, Appellant's contend that the proposed *Leung-Rumbaugh* combination does not disclose, teach, or suggest "the data is or represents a X.509 certificate," as recited in Appellant's Claim 4. In the *Office Action*, the Examiner relies upon Figure 2 and page 737, column 2, paragraph 5 of *Leung*, specifically, for disclosure of the recited claim elements. Appellant respectfully disagrees.

Figure 2 of *Leung* merely depicts the functional model of a directory system. As illustrated in Figure 2, the system includes multiple directory users that communicated with Directory System Agents (DSA) through a Directory User Agent (DUA). As described in the cited portion of *Leung*, a Directory Service Element Processor (DSEP), "which is the front-end processor, is responsible for communicating with DUAs through the use of an OSI communication stack." (*Leung*, Page 737, column 2, paragraph 5). "Requests, which are coded and sent using the DAP protocol, are sent from a DUA to DSEP by the communication stack." (*Leung*, Page 737, column 2, paragraph 5). Thus, *Leung* only discloses the system components used to send/receive a coded request. *Leung* does not disclose that the data stored in the directory system is or represents a X.509 certificate. These elements are absent from *Leung*. Accordingly, *Leung* does not disclose, teach, or suggest "the data is or represents a X.509 certificate," as recited in Appellant's Claim 4.

B. Conclusions with Respect to Claim 4

For at least these reasons, Appellants respectfully submit that the Examiner has not established a *prima facie* case of obviousness based on the proposed *Leung-Rumbaugh* combination with respect to independent Claim 4. Thus, for at least these reasons, Appellants submit that these rejections are improper and respectfully request that the Board reverse these rejections of independent Claim 4.

V. Claim 5 is Allowable over the proposed *Leung-Rumbaugh* Combination

Claim 5 is patentable at least because it depends on Claim 1, which Appellant has shown above to be allowable. Additionally, Claim 5 recites features that are distinguishable from the prior art. For example, Claim 5 recites that "a selected one of the data components is a checksum or fingerprint." Appellants will demonstrate below that this combination of features and operations is not disclosed, taught, or suggested in the proposed *Leung-Rumbaugh* combination.

A. The proposed Leung-Rumbaugh combination does not disclose, teach, or suggest "*the data is or represents a X.509 certificate*"

As an example of the deficiencies of the cited references, Appellant's contend that the proposed *Leung-Rumbaugh* combination does not disclose, teach, or suggest "a selected one of the data components is a checksum or fingerprint," as recited in Appellant's Claim 5. In the *Office Action*, the Examiner relies upon page 738, column 1, paragraph 1 of *Leung*, specifically, for disclosure of the recited claim elements. Appellant respectfully disagrees.

The cited portion of *Leung* merely discloses that a Directory Operator Processor (DOP) "organizes the DIBP requests in a logical sequence, sends them to DIBP, and collates the corresponding results returned by DIBP." (*Leung*, Page 738, column 1, paragraph 1). "After collecting the results, [the DOP] passes them to DSEP in a form that DSEP understands." (*Leung*, Page 738, column 1, paragraph 1). Thus, the cited portion of *Leung* only discloses the collection and collating of results. *Leung* does not disclose that the data stored in a table and/or that the data identifiable by searching is a checksum or fingerprint. These elements are absent from *Leung*. Accordingly, *Leung* does not disclose, teach, or suggest "a selected one of the data components is a checksum or fingerprint," as recited in Appellant's Claim 5.

B. Conclusions with Respect to Claim 5

For at least these reasons, Appellants respectfully submit that the Examiner has not established a *prima facie* case of obviousness based on the proposed *Leung-Rumbaugh* combination with respect to independent Claim 5. Thus, for at least these reasons, Appellants

submit that these rejections are improper and respectfully request that the Board reverse these rejections of independent Claim 5.

VI. Claims 25-28 are Allowable over the proposed *Leung-Rumbaugh* Combination

Claims 25-28 are patentable at least because they depend on Claims 22, which Appellant has shown above to be allowable. Additionally, Claims 25-28 recite additional features that are distinguishable from the art. For example, Claim 25 recites that "the data is or represents one or more of the following: a X.509 certificate, a checksum of the data, and a fingerprint of the data." Likewise, Claim 23 recites that the component is a checksum or fingerprint of the data. Claims 27 and 28 recite further operations and features where the data is a fingerprint or checksum.

In the *Office Action*, the Examiner continues to rely upon *Leung* for disclosure of these elements. (*Final Office* Action, pages 7-9). Appellants have shown above with regard to Claims 4 and 5, however, that *Leung* does not disclose, teach, or suggest that the data is or represents a X.509 certificate, a checksum of the data, or a fingerprint of the data. The cited portions of *Leung* only discloses that the system components used to send/receive a coded request and that the results are collected and collated. (*Leung*, Page 737, column 2, paragraph 5; Page 738, column 1, paragraph 1). Accordingly, for reasons similar to those discussed above with regard to Claims 4 and 5, *Leung* does not disclose that the data stored in the directory system is or represents a X.509 certificate, a checksum, or a fingerprint. These elements are absent from *Leung*. Accordingly, *Leung* does not disclose, teach, or suggest that "the data is or represents one or more of the following: a X.509 certificate, a checksum of the data, and a fingerprint of the data," as recited in Claim 25. *Leung* similarly does not disclose, teach, or suggest that "the component is a checksum or fingerprint of the data," as recited in Claim 26. *Leung* also does not disclose, teach, or suggest that "the search is conducted using a search table to locate the fingerprint or checksum," as recited in Claim 27, or that "components of the checksum or fingerprint are searched," as recited in Claim 28.

For at least these reasons, Appellants respectfully submit that the Examiner has not established a *prima facie* case of obviousness based on the proposed *Leung-Rumbaugh* combination with respect to independent Claims 25-28. Thus, for at least these reasons,

Appellants submit that these rejections are improper and respectfully request that the Board reverse these rejections of independent Claims 25-28.

VII. The Proposed Combinations are Improper.

The legal standard for establishing a *prima facie* case of obviousness based on modification or combination of prior art references is set out above in Section I of the Arguments section of this Appeal Brief. Briefly, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings. M.P.E.P. § 2142, 2143. Even the fact that references *can* be modified or combined does not render the resultant modification or combination obvious unless the prior art teaches or suggests the desirability of the modification or combination. The required evidence of such a teaching, suggestion, or motivation is essential to avoid impermissible hindsight reconstruction of an applicant's invention. *In re Dembiczak*, 175 F.3d at 999, 50 U.S.P.Q.2d at 1617.

In the *Final Office Action*, the Examiner acknowledges, with regard to independent Claim 1, for example, that *Leung* does not disclose creating a second table storing data components and having one row for each component of the data. (*Office Action*, page 5). To maintain the rejection, the Examiner speculates:

> It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method of Leung by creating a second table storing data components and having one row for each component of the data as disclosed by Rumbaugh (see Rumbaugh Fig. 17.2, page 370, paragraph 17.2.3 and Fig. 17.16) . . . to provide an excellent basis for modeling object oriented data base management system (DBMS) (see Rumbaugh page 388, paragraph 17.5), therefore, improving the performance of the directory searching methods and system.

(*Office Action*, page 6). Appellant respectfully disagrees.

It appears that the Examiner has merely proposed alleged advantages of combining *Leung* with *Rumbaugh* (advantages which Applicant does not admit could even be achieved by combining these references in the manner the Examiner proposes). While the Examiner has cited a portion of *Rumbaugh* that touts an advantage of its techniques, the cited advantage doe not explain why one of ordinary skill in the art would have been motivated to combine

the DIT and ENTRY tables disclosed in *Leung* with the method for translating an object model into a table model as disclosed in *Rumbaugh*. Specifically, the alleged advantage of the system disclosed in *Rumbaugh* does not provide an explanation as to: (1) why it would have been obvious to one of ordinary skill in the art at the time of Applicant's invention (***without using Applicant's claims as a guide***) to modify the particular techniques disclosed in *Leung* with the cited disclosure of *Rumbaugh*; (2) how one of ordinary skill in the art at the time of Applicant's invention would have actually done so; and (3) how doing so would purportedly meet the limitations of Applicant's claims.

Indeed, if it were sufficient for Examiners to merely point to a purported advantage of one reference and conclude that it would have been obvious to combine of modify that reference with other references simply based on that advantage (which, as should be evident from the case law discussed above, it certainly is not), then virtually any two or more references would be combinable just based on the fact the one reference states an advantage of its system. Of course, as the Federal Circuit has made clear and as discussed above, that is not the law. Accordingly, Applicant respectfully submits that the Examiner's conclusions set forth in the Office Action do not meet the requirements set forth in the M.P.E.P. and the governing Federal Circuit case law for demonstrating a *prima facie* case of obviousness.

Furthermore, it is improper for an Examiner to use hindsight having read the Applicant's disclosure to arrive at an obviousness rejection. *In re* Fine, 837 F.2d 1071, 1075, 5 U.S.P.Q.2d 1596, 1600 (Fed. Cir. 1988). It is improper to use the claimed invention as an instruction manual or template to piece together the teachings of the prior art so that the claimed invention is rendered obvious. *In re* Fritch, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). It is clear based at least on the many distinctions discussed above that the proposed *Leung-Rumbaugh* combination does not, taken as a whole, suggest the claimed invention, taken as a whole. Rather, Applicant respectfully submits that the Examiner has merely pieced together disjointed portions of references, with the benefit of hindsight using Applicant's claims as a blueprint, in an attempt to reconstruct Applicant's claims.

For at least these reasons, Applicant submits that the rejection of Claims 1-28 is improper. Applicant respectfully requests reconsideration and allowance of Claims 1-28.

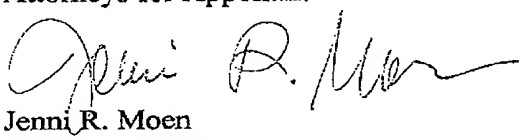
Conclusion

Appellant has demonstrated that the present invention, as claimed, is clearly distinguishable over the prior art cited by the Examiner. Therefore, Appellant respectfully requests the Board to reverse the final rejections and instruct the Examiner to issue a Notice of Allowance with respect to all pending claims.

The Commissioner is hereby authorized to charge $510.00 for filing this Brief in support of an Appeal to Deposit Account No. 02-0384 of Baker Botts, L.L.P. No other fees are believed due; however, the Commissioner is authorized to charge any additional fees or credits to Deposit Account No. 02-0384 of Baker Botts, L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Appellant



Jenni R. Moen
Reg. No. 52,038
(214) 953-6809

Dated: July 18, 2008

Correspondence Address:

at Customer No. **05073**

APPENDIX A

Pending Claims

1. A method of arranging and searching for data in a database comprising:

creating a first table storing data comprising at least one data entry, the data entry comprising a plurality of data components, the first table comprising one row for each data entry;

creating a second table storing the plurality of data components of the data entry of the first table, the second table comprising one row for each of the plurality of data components of the data entry of the first table; and

searching the rows of the second table to identify a particular one of the plurality of data components; and

returning the given data entry from the first table that includes the particular one of the plurality of data components.

2. The method as claimed in claim 1, wherein the data is a structured data type.

3. The method as claimed in claim 1, wherein the data is a string data type.

4. The method as claimed in claim 1, wherein the data is or represents a X.509 certificate.

5. The method as claimed in claim 1, wherein a selected one of the data components is a checksum or fingerprint.

6. The method as claimed in claim 1, where the database is a part of an electronic directory services system.

7. The method as claimed in claim 6, where the electronic directory services system comprises an X.500 and LDAP services system.

22. A method of searching a database for given data entries, comprising:

determining a component of a given data entry of a first table, the given data entry of the first table comprising a plurality of data components;

identifying a component identifier indicating a data type that is associated with the component of the first table;

using the component identifier indicating the data type to execute one of an exact or initial matching on a column of a second table in order to locate the component in the second table, the second table comprising one row for each of the plurality of data components of the given data entry of the first table; and

returning the given data entry from the first table matching the component located.

23. The method as claimed in claim 22, where the database is a part of an electronic directory services system.

24. The method as claimed in claim 22, where the electronic directory services system comprises an X.500 and LDAP services system.

25. The method as claimed in claim 22, wherein the data is or represents one or more of the following: a X.509 certificate, a check sum of the data, and a fingerprint of the data.

26. The method as claimed in claim 23, wherein the component is a checksum or fingerprint of the data.

27. The method as claimed in claim 26, wherein the search is conducted using a search table to locate the fingerprint or checksum.

28. A method as claimed in claim 27, further wherein components of the checksum or fingerprint are searched.

APPENDIX B

Evidence Appendix

No evidence was submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132, and no other evidence was entered by the Examiner and relied upon by Appellant in the Appeal.

APPENDIX C

Related Proceedings Appendix

As stated on Page 3 of this Appeal Brief, to the knowledge of Appellant's Counsel, there are no known appeals, interferences, or judicial proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision regarding this Appeal.